

APPENDIX VI - ERROR EVENT

© Copyright 2003 Time Warner Cable, Inc. All rights reserved.

```
import java.lang.*;
import java.util.*;

/**
 */
public class ErrorEvent implements IMessageEvent
{
    public final static int INFO_SHUTDOWN_IMMINENT = 0x0;

    public final static int ERROR_UNCAUGHT_JAVA_EXCEPTION = 0X30000001;

    // Static error event Id number. Incremented for each new event.
    private static long lastEventId;

    private long eventId;
    private long appId;
    private int typeCode;
    private long date;
    private String message;
    private Throwable errorOrException;

    /**
     * Three argument ErrorEvent constructor.
     *
     * @param typeCodeIn - The value indicating the error; see IMessageEvent.
     *
     * @param errorMsg - The readable error message in the following format:
     */
    public ErrorEvent(int typeCodeIn, long appIdIn, String messageIn)
    {
        this(typeCodeIn, appIdIn, messageIn, null);
    }

    /**
     * Four argument ErrorEvent constructor.
     *
     * @param errorType - The value indicating the error; see IMessageEvent.
     *
     * @param errorMsg - The readable error message in the following format:
     */
    public ErrorEvent(int typeCodeIn,
                      long appIdIn,
                      String messageIn,
                      Throwable errorOrExceptionIn)
    {
        lastEventId = (lastEventId == Long.MAX_VALUE) ? 0 : lastEventId + 1;
        eventId = lastEventId;
        typeCode = typeCodeIn;
        date = System.currentTimeMillis();
        message = messageIn;
        errorOrException = errorOrExceptionIn;
    }

    /**
```

```

    * Get the unique identifier of the event.
    *
    * @return unique identifier of the event.
    */
5  public long getEventId()
    {
        return eventId;
    }

10  /**
    * Get the unique identifier of the application that logged the event.
    *
    * @return The application identifier.
    */
15  public long getAppId()
    {
        return appId;
    }

20  /**
    * Get the event type code. Identifies a code specific to the event system.
    *
    * @return type code of the event.
    */
25  public int getTypeCode()
    {
        return typeCode;
    }

30  /**
    * Set the error type code. Allows an application to change the error received.
    *
    * @param error - The new error value.
    */
35  public void setTypeCode(int typeCodeIn)
    {
        typeCode = typeCodeIn;
    }

40  /**
    * Get the event date in milli-seconds from midnight January 1, 1970.
    *
    * @return Date the event was submitted to the system.
    */
45  public long getDate()
    {
        return date;
    }

50  /**
    * Set the date. Allows an application to change the date received.
    *
    * @param error - The new error value.
    */
55  public void setDate(long dateIn)
    {
        date = dateIn;
    }

```

```

    }

    /**
    * Get the readable message.
    5    *
    * @return the readable event message.
    */
    public String getMessage()
    {
    10    return message;
    }

    /**
    * Set the error message string. Allows an application to change the error message received.
    15    *
    * @param error - The new error message.
    */
    public void setErrorMessage(String messageIn)
    {
    20    message = messageIn;
    }

    protected void finalize()
    {
    25    // Store the lastErrorEventId in a persistent storage object.
    }

    static {
    30    if(/*some persistent storage object exists with error event Id stored in it*/ true)
        lastEventId = /* value in the persistent storage object */1;
    else
        lastEventId = 0;
    }
    35 }

```